

Fig. 1

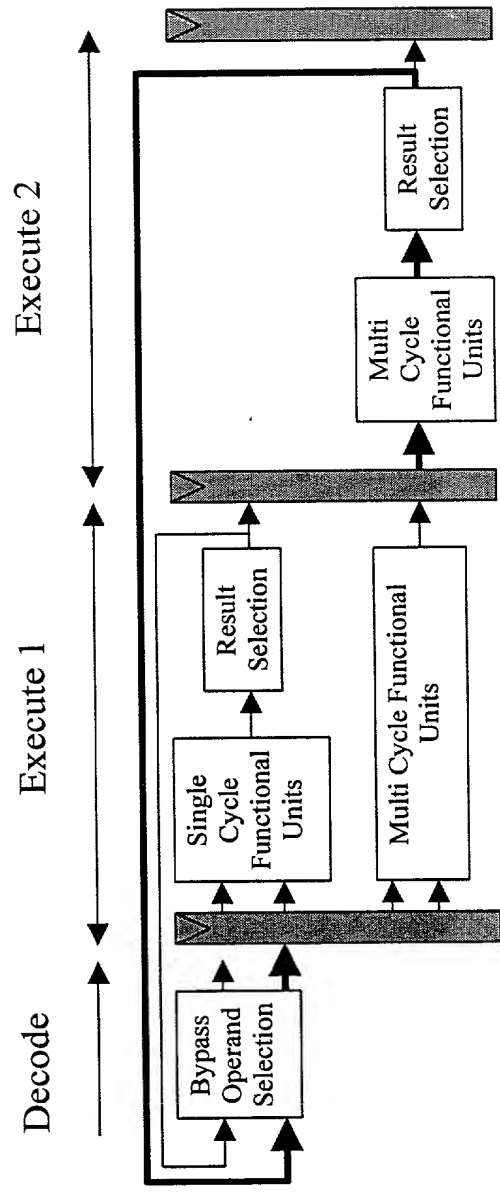


Fig. 2

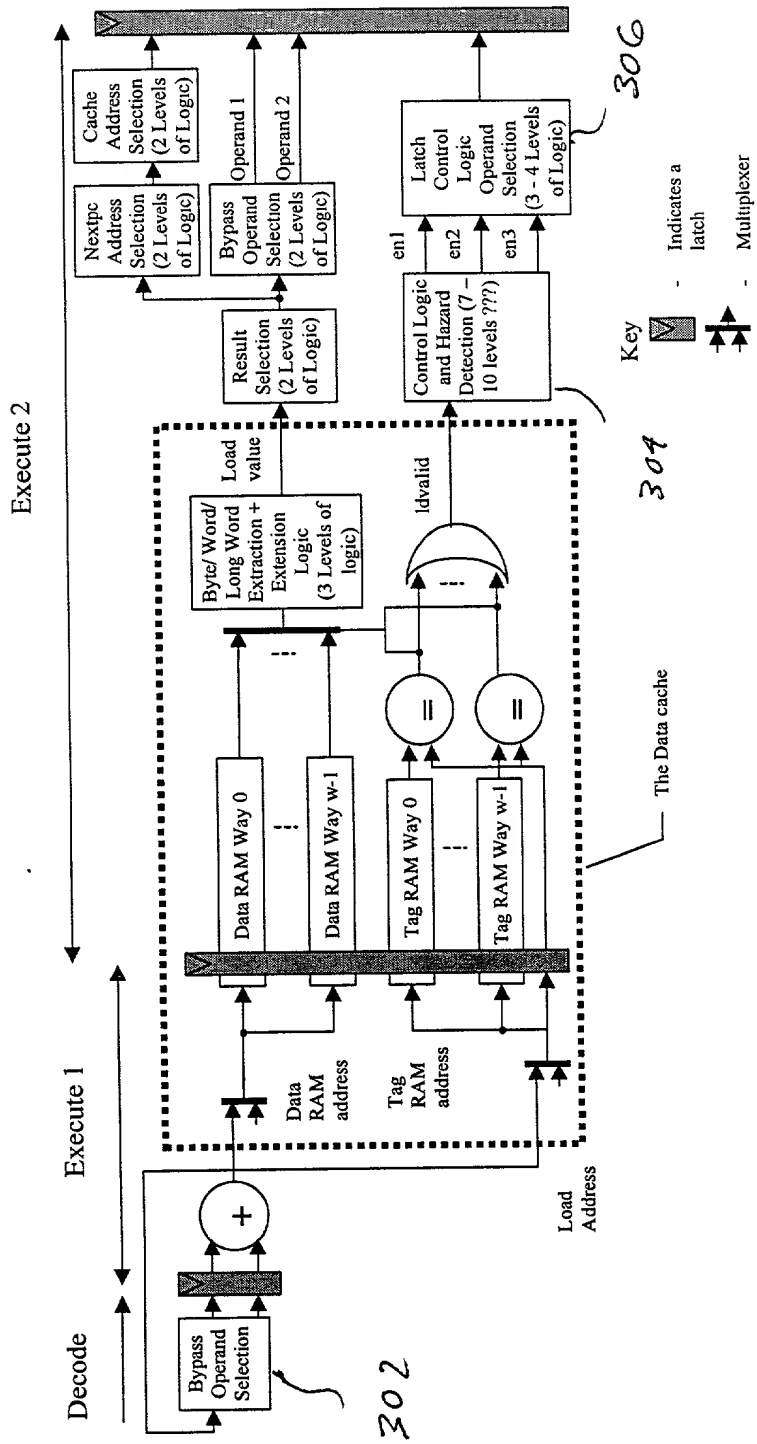


Fig. 3

**Fig. 3a.**

Pipeline Stage						
Reference Step	0	F	D	E1	E2	WB
350	Ld					
352	Mov	Ld				
354	Add	Mov	Ld			
356		Add	Mov	Ld		
358			Add	Mov	Ld	
360			Add		Mov/Ld	
362			Add		Ld	Mov
364				Add		Ld
366					Add	
368						Add

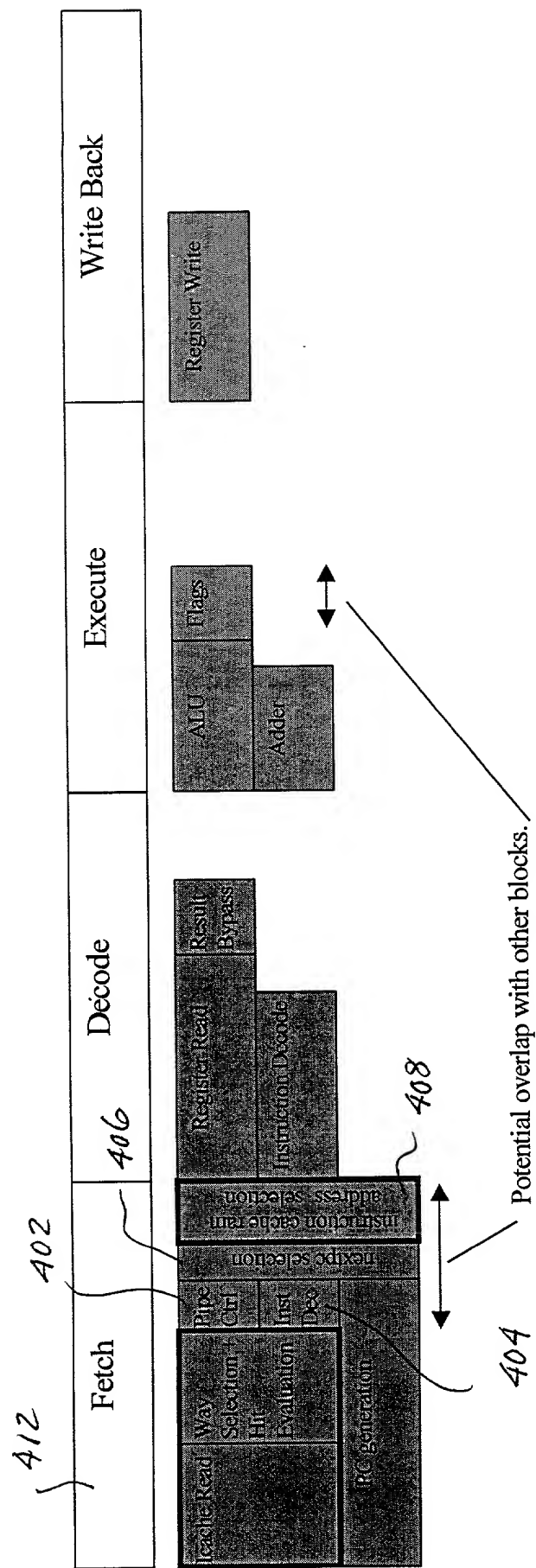


Fig. 4

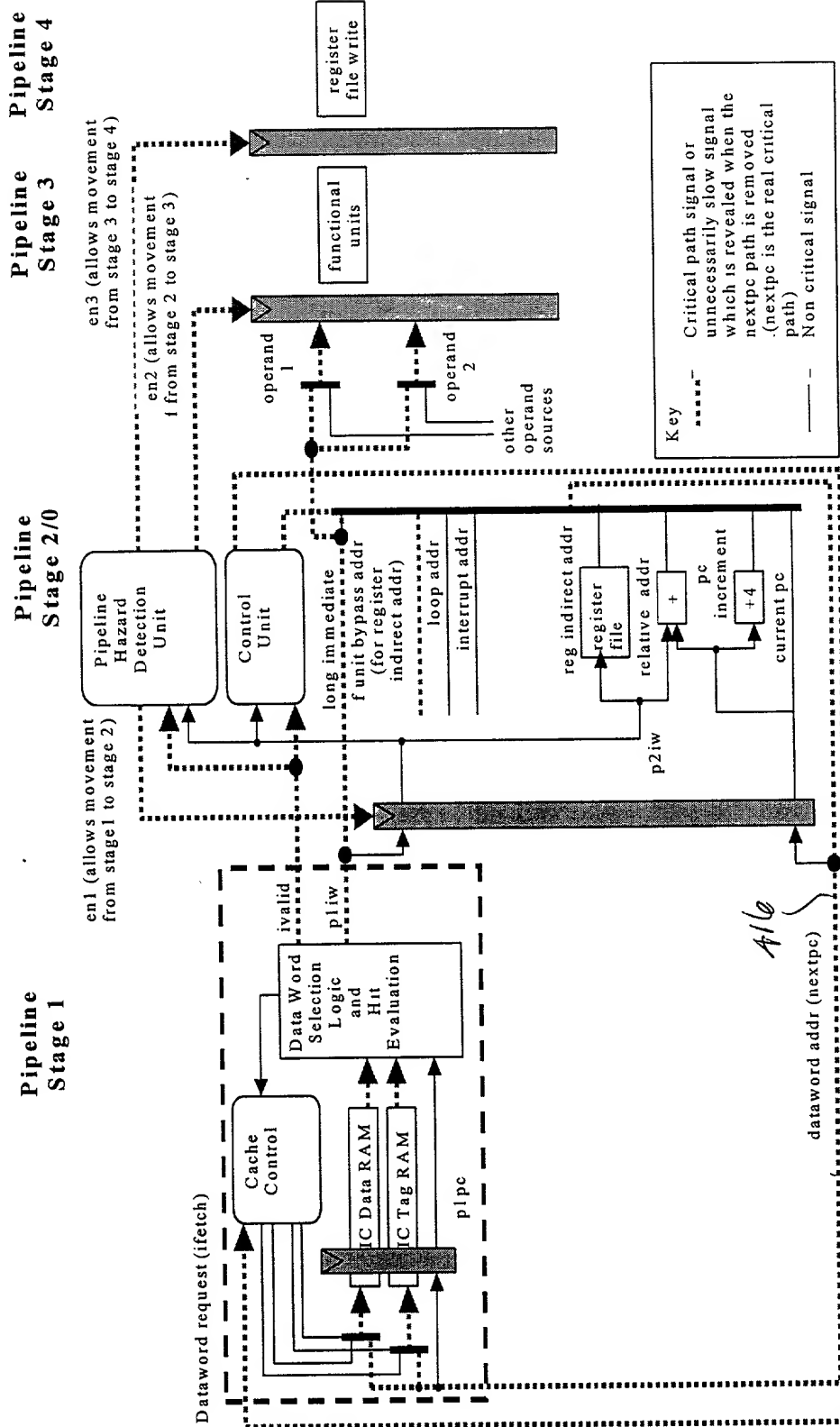


Fig. 4a

FIG. 5

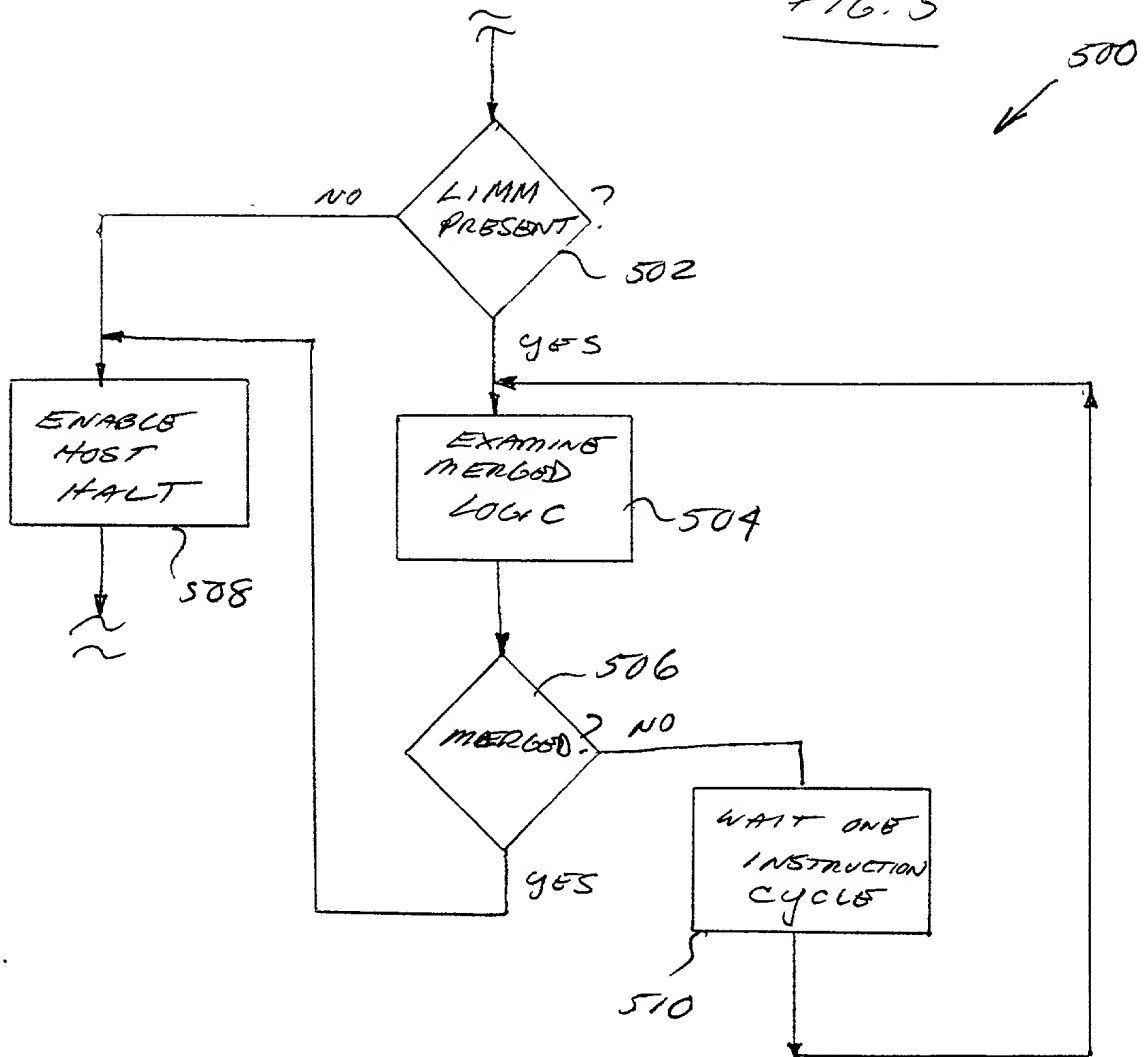


FIG. 6 is a block diagram of a processor pipeline. The pipeline is divided into four stages: Fetch, Decode, Execute, and Write Back. The Fetch stage includes a PC Read block (602) and a Branch Prediction block (604). The Decode stage includes a Register Read block (610) and an Instruction Decode block (610). The Execute stage includes an ALU block (606) and a Result Bypass block (606). The Write Back stage includes a Register Write block (606).

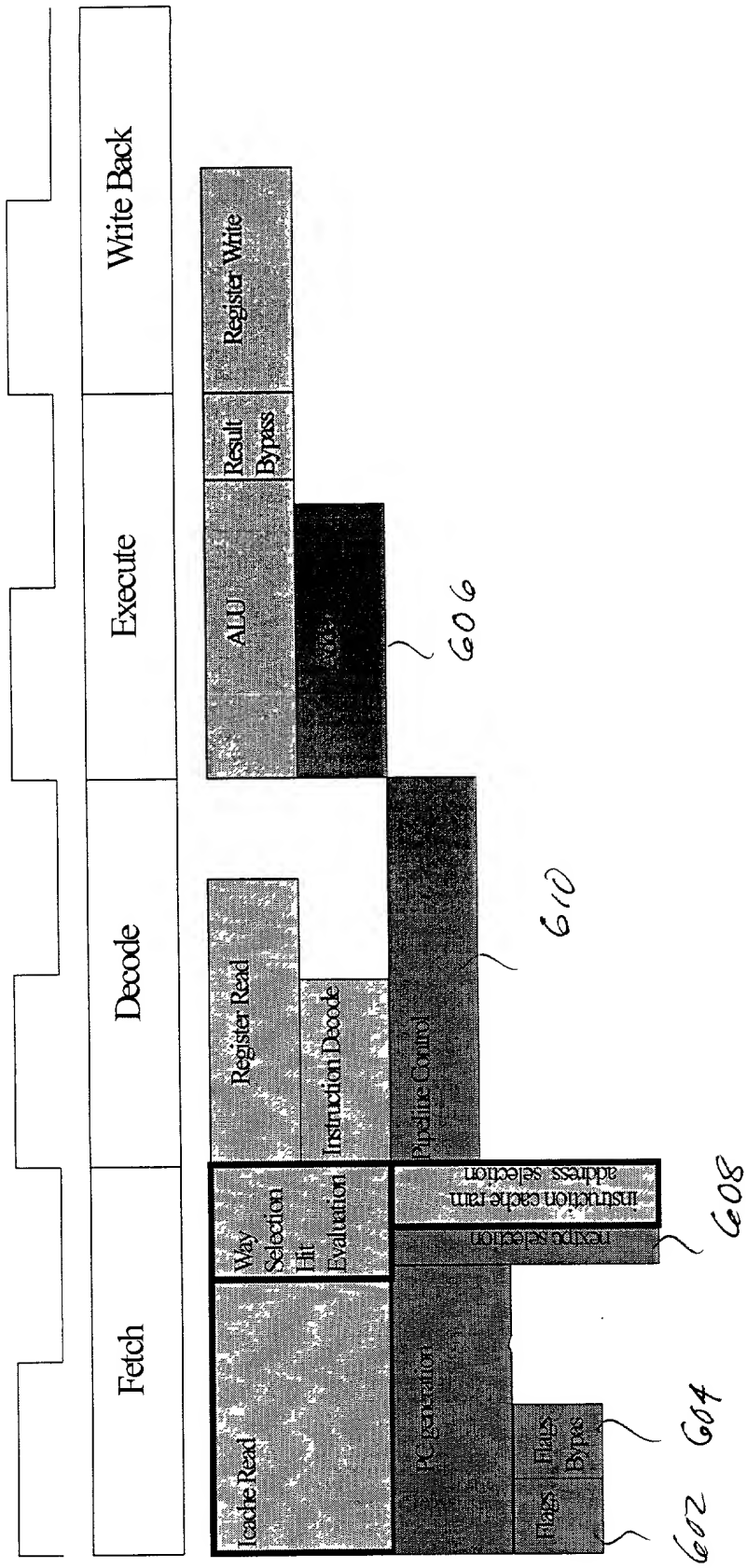


Fig. 6



FIG. 7 is a block diagram of a four-stage pipeline processor 700. The processor includes a Pipeline Stage 1 (702), Pipeline Stage 2/0 (722), Pipeline Stage 3 (724), and Pipeline Stage 4 (730). Pipeline Stage 1 includes a Cache Control (704), IC Data RAM (706), IC Tag RAM (708), and Data Word Selection Logic and Hit Evaluation (710). Pipeline Stage 2/0 includes a Pipeline Hazard Detection Unit (722) and a Control Unit (724). Pipeline Stage 3 includes functional units (724). Pipeline Stage 4 includes a register file write (730). The processor also includes a register file read (732), a pc increment (+4) (734), and a current pc (736). The processor is controlled by a dataport request (ifetch) (700) and a dataport address (nextpc) (700). The processor is also controlled by a dataport address (nextpc) (700). The processor is also controlled by a dataport address (nextpc) (700).

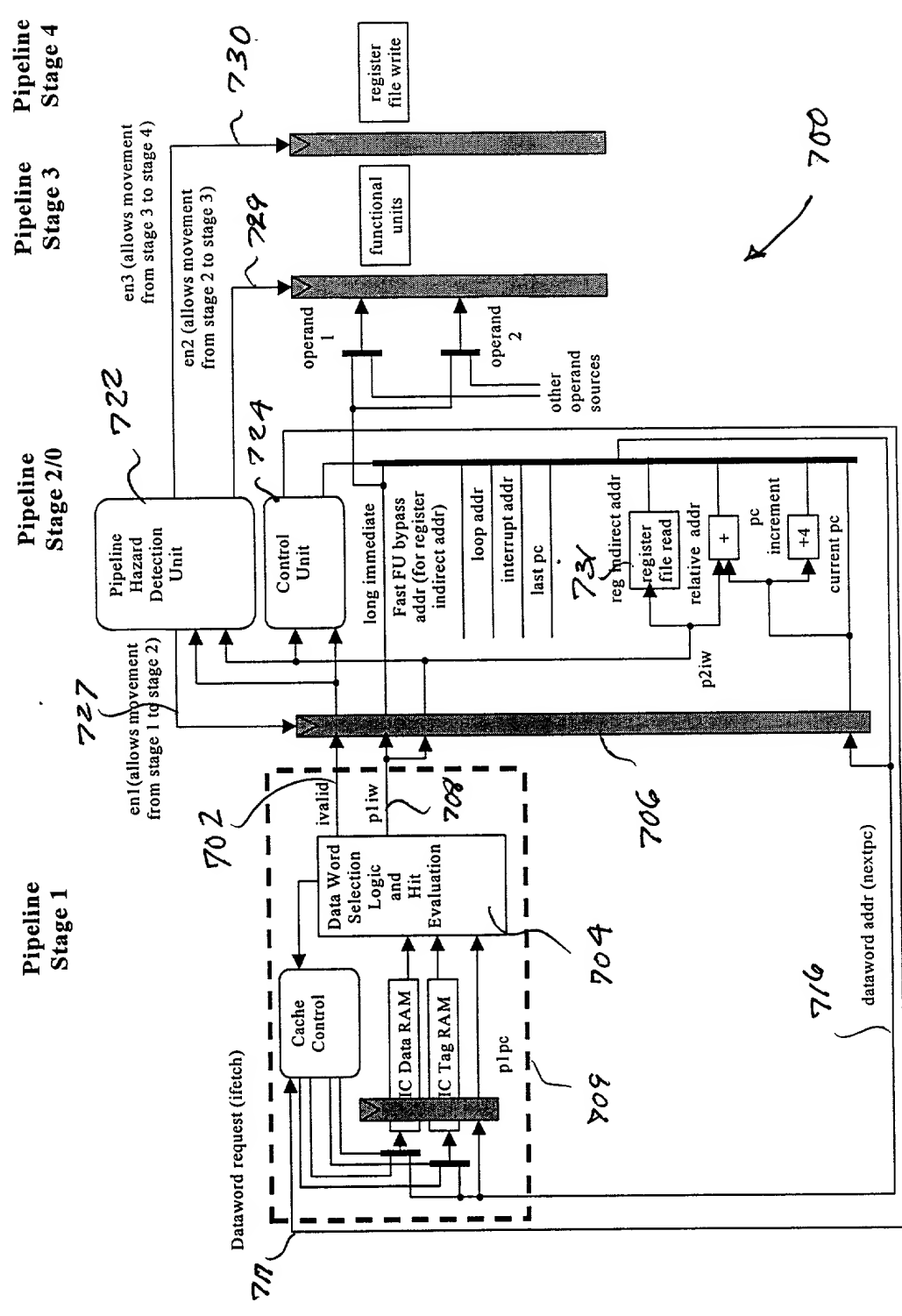


Fig. 7

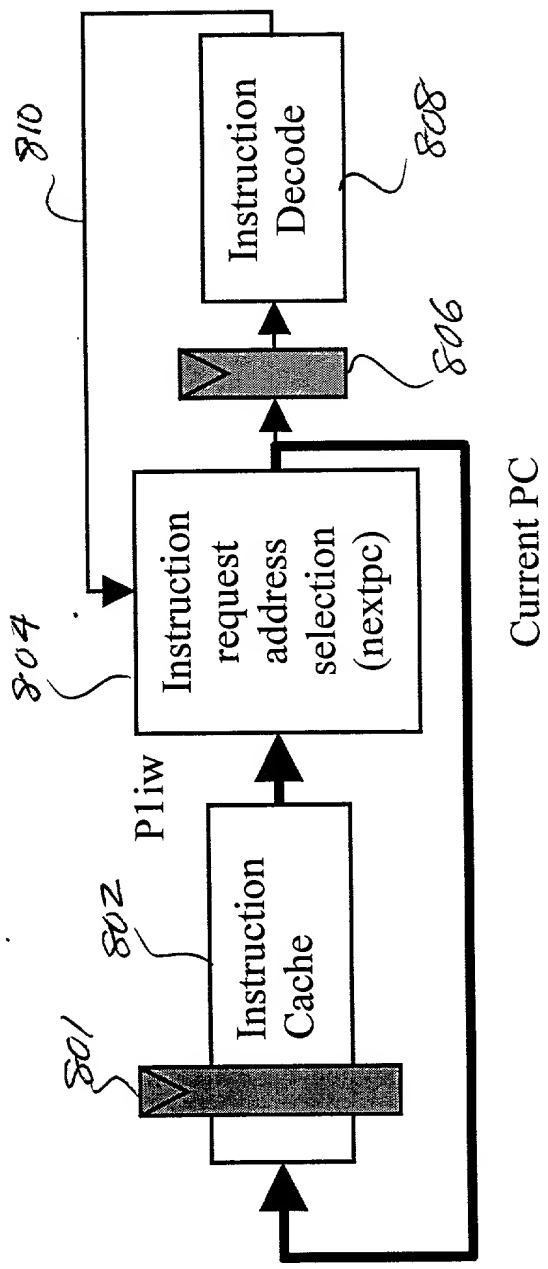


Fig. 8

800

**Fig. 8a**

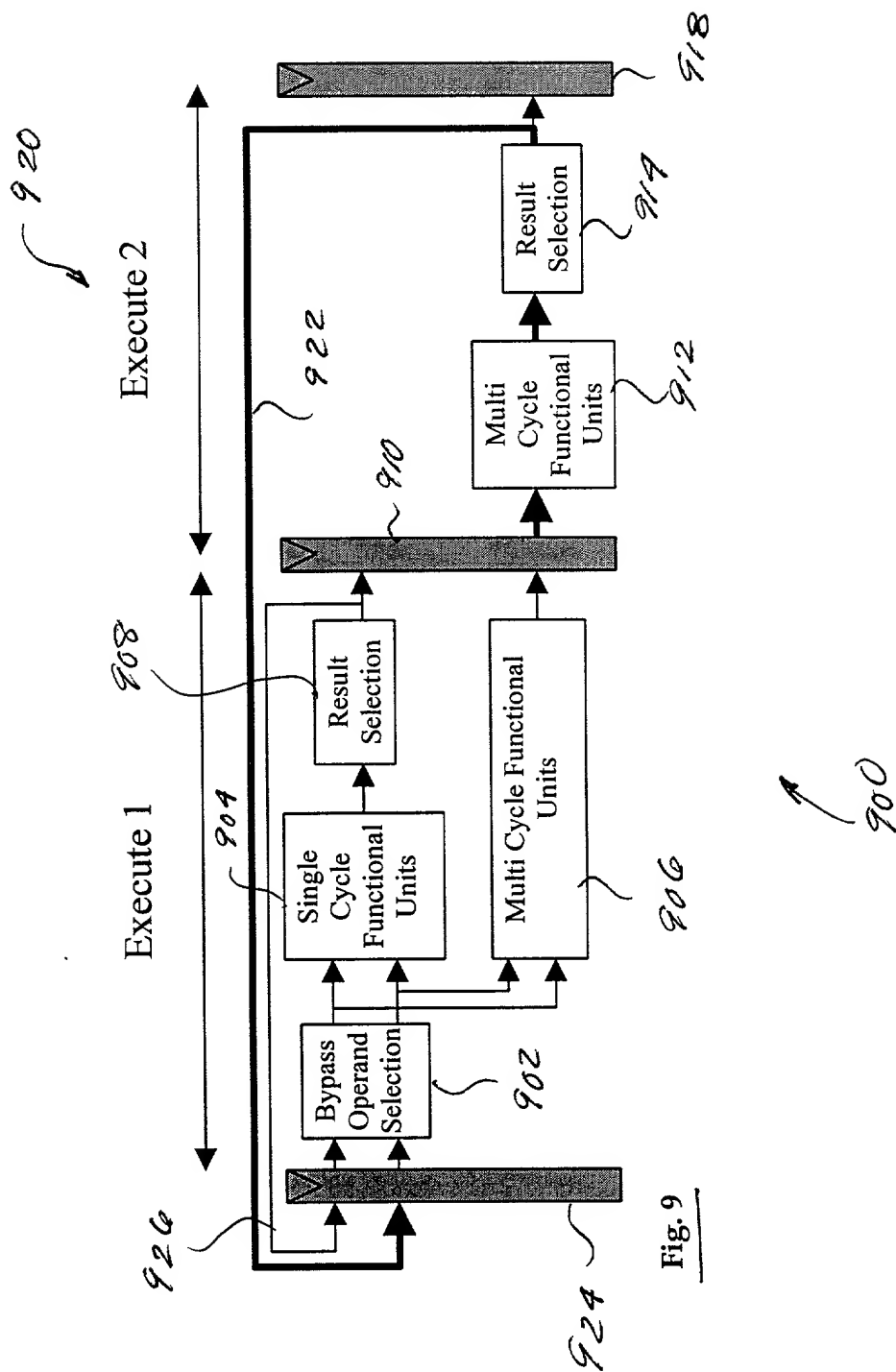
Reference Step	Instruction Fetch Address (Inst Addr: next_pc)	Pipeline Stage			
		Fetch (Inst Addr: current_pc)	Decode (Inst Addr: last_pc)	Execute	Write- back
820	J.d <sub>A</sub>				
822	Brk <sub>B</sub>	J.d <sub>A</sub>			
824	Target <sub>C</sub>	Brk <sub>B</sub>	J.d <sub>A</sub>		
826		Target <sub>C</sub>	Brk <sub>B</sub>	J.d <sub>A</sub>	
828			Brk		J.d <sub>A</sub>
			Restart		
830			Brk		
			Restart		
832	Add <sub>B</sub>		Brk		
			Restart		
834	Target <sub>C</sub>	Add <sub>B</sub>			
836	Target <sub>2C</sub>	Target <sub>C</sub>	Add <sub>B</sub>		
838	Target <sub>3C</sub>	Target <sub>2C</sub>	Target <sub>C</sub>	Add <sub>B</sub>	

**Fig. 8b**

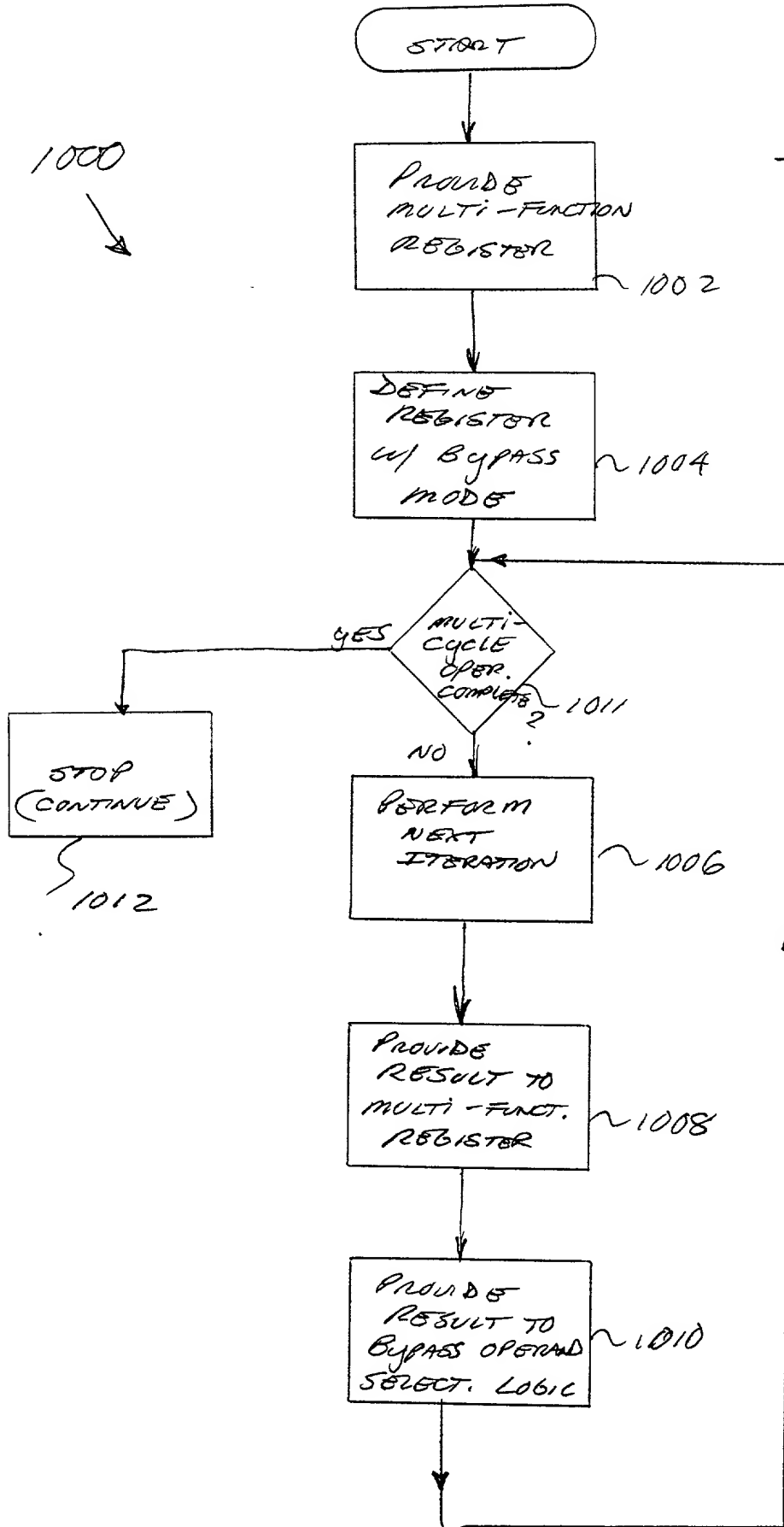
Reference Step	Instruction Fetch Address (Inst Addr: next_pc)	Pipeline Stage			
		Fetch (Inst Addr: currentpc)	Decode (Inst Addr: last_pc)	Execute	Write-back
840	Add <sub>A</sub>				
842	Brk <sub>B</sub>	Add <sub>A</sub>			
844	Mov <sub>C</sub>	Brk <sub>B</sub>	Add <sub>A</sub>		
846		Mov <sub>C</sub>	Brk <sub>B</sub>	Add <sub>A</sub>	
848			Brk Restart		Add <sub>A</sub>
850			Brk Restart		
852	Add <sub>B</sub>		Brk Restart		
854	Mov <sub>C</sub>	Add <sub>B</sub>			
856	Mov2 <sub>C</sub>	Mov <sub>C</sub>	Add <sub>B</sub>		
858	Mov3 <sub>C</sub>	Mov2 <sub>C</sub>	Mov <sub>C</sub>	Add <sub>B</sub>	

**Fig. 8c**

Reference	Instruction Fetch Address (Inst Addr: next_pc)	Pipeline Stage			
		F (Inst Addr: currentpc)	D (Inst Addr: last_pc)	E	W
860	J.d <sub>A</sub>				
862	Brk <sub>B</sub>	J.d <sub>A</sub>			
864	Target <sub>C</sub>	Brk <sub>B</sub>	J.d <sub>A</sub>		
866		Target <sub>C</sub>	Brk <sub>B</sub>	J.d <sub>A</sub>	
868			Brk		
870			Restart		
			Brk	J.d <sub>A</sub>	
872	Add <sub>B</sub>		Restart		
			Brk	J.d <sub>A</sub>	
874	Target <sub>C</sub>	Add <sub>B</sub>		J.d <sub>A</sub>	
876	Target2 <sub>C</sub>	Target <sub>C</sub>	Add <sub>B</sub>	J.d <sub>A</sub>	
878	Target3 <sub>C</sub>	Target2 <sub>C</sub>	Target <sub>C</sub>	Add <sub>B</sub>	J.d <sub>A</sub>



F76.10



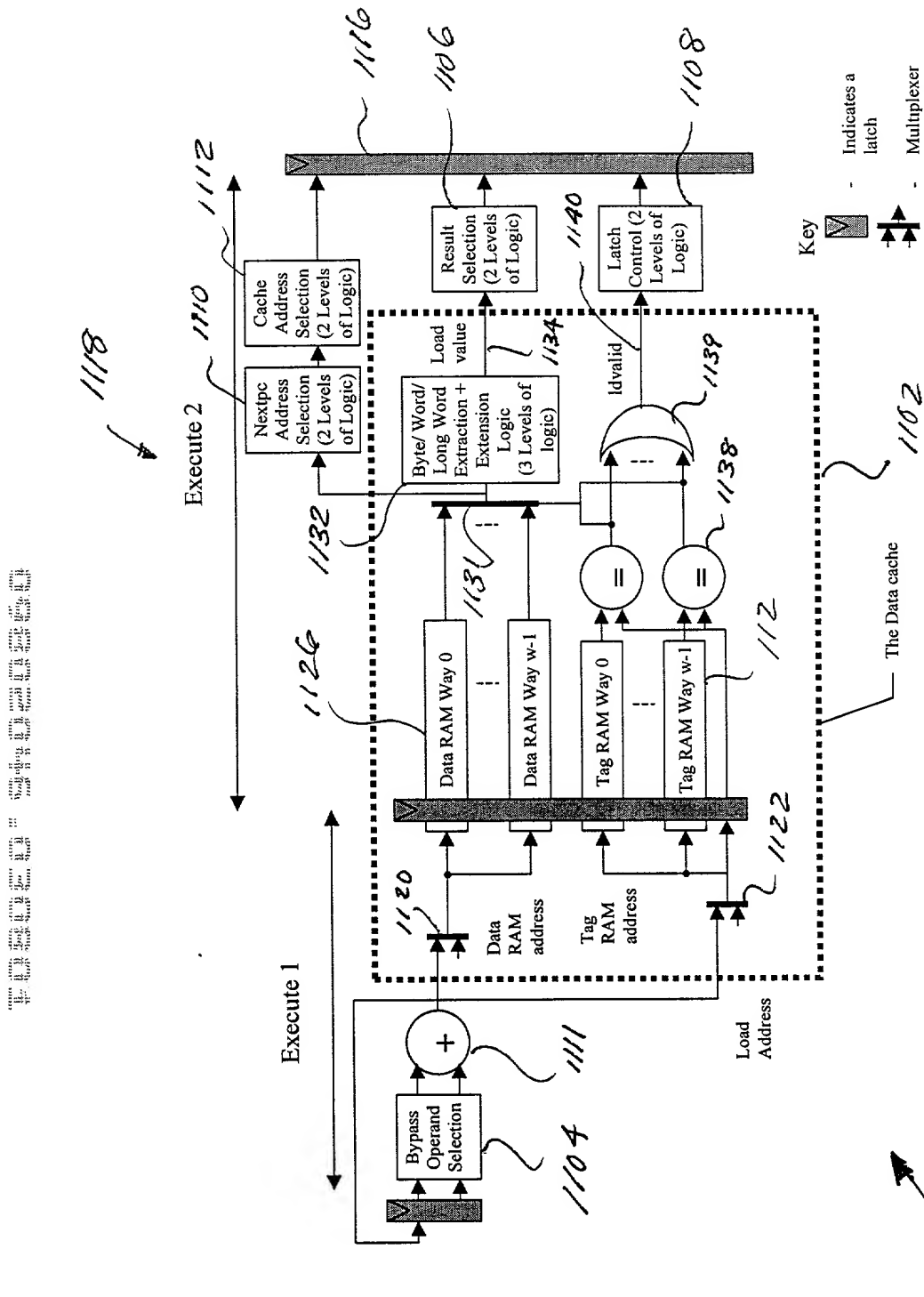
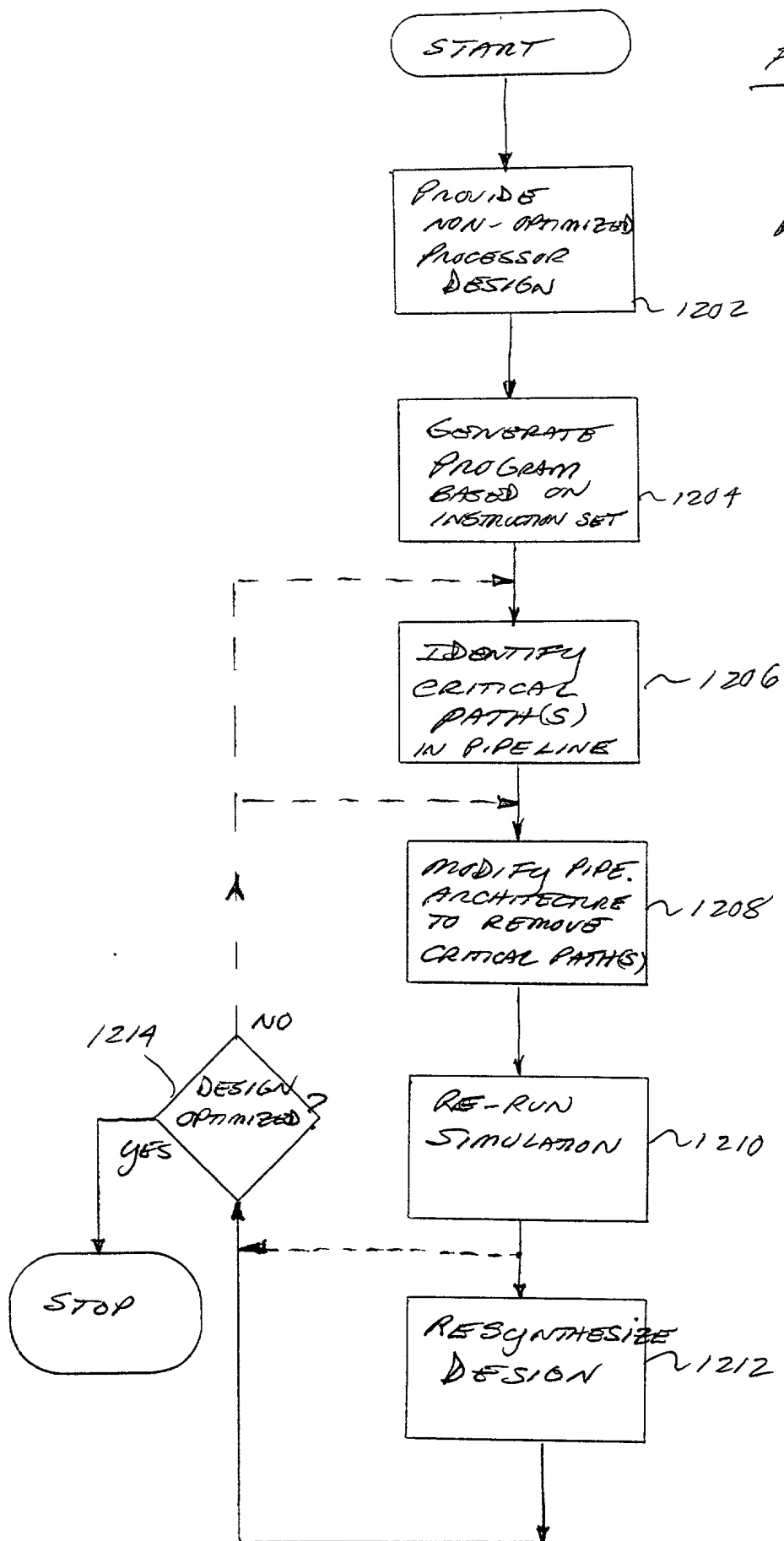




Fig. 11a

Pipeline Stage						
Reference	0	F	D	E1	E2	WB
1174	Ld					
1176	Mov	Ld				
1178	Add	Mov	Ld			
1180		Add	Mov	Ld		
1182			Add	Mov	Ld	
1184				Add	Mov/Ld	
1186				Add	Ld	Mov
1188				Add		Ld
1190				Add	Add	
1192						Add

FIG. 12



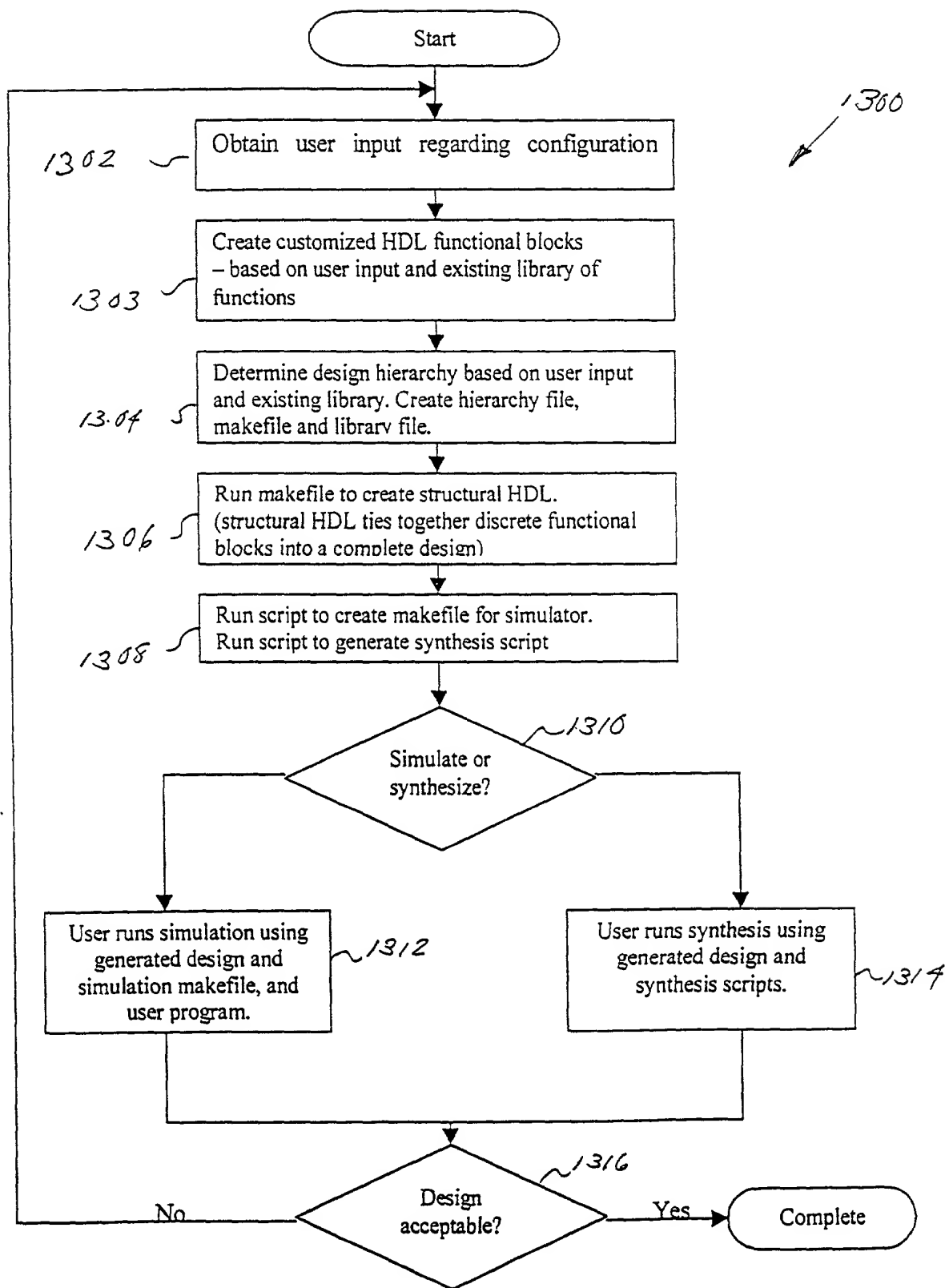


FIG. 13

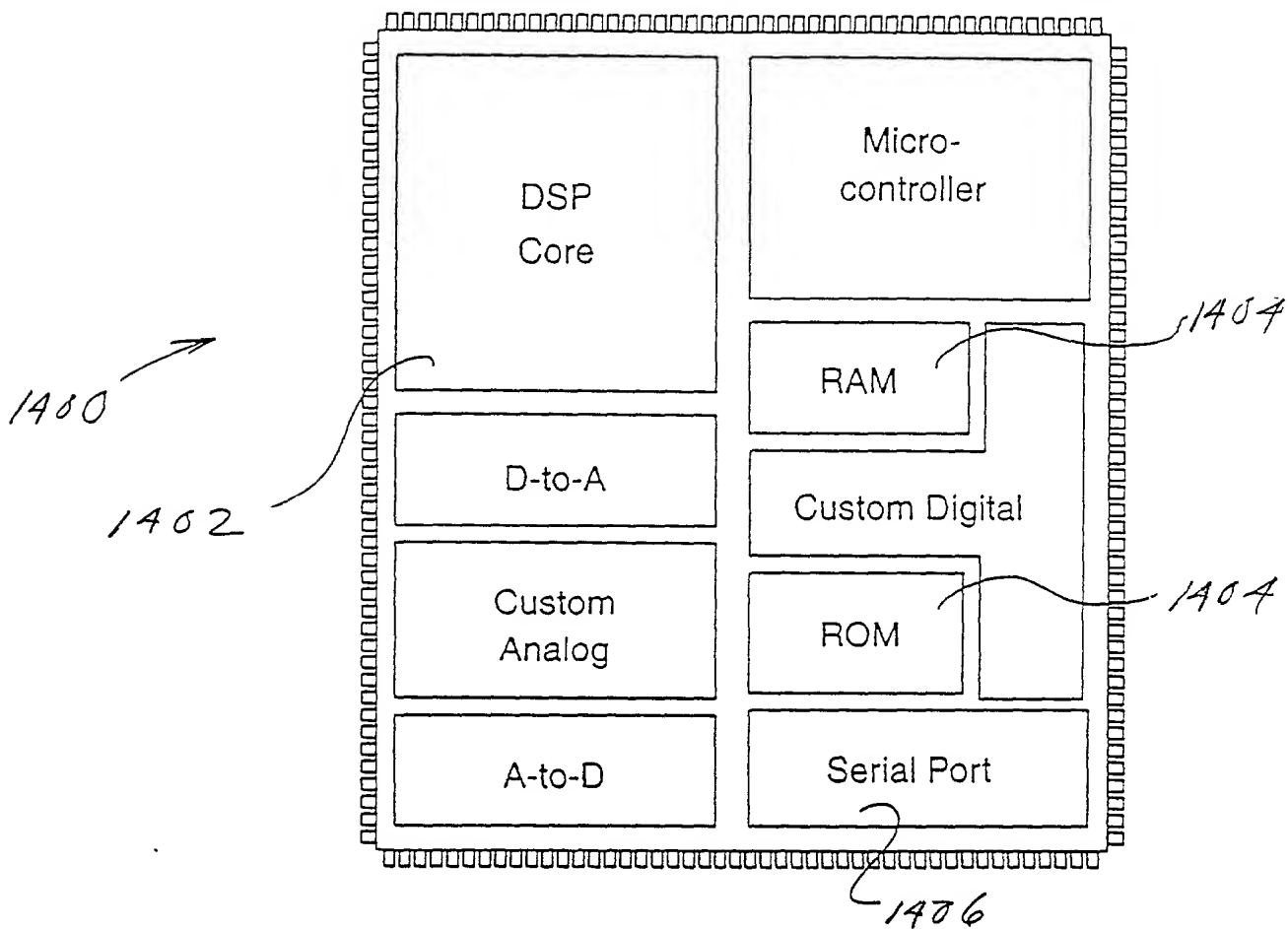


FIG. 1A

[illegible]